Machine Learning 1

Lecture 8.2 - Supervised Learning
Neural Networks - Universal Approximators

Erik Bekkers

(Bishop 5.1)

Slide credits: Rianne van den Berg

Image credit: Kirillm | Getty Images

NN: Universal Approximators

Theorem: Universal Approximators

- Let f be any continuous function on a compact area of $\mathbb{R}^D$

  activation fn

- Let h any fixed analytic function which is not polynomial (e.g. logistic function, tanh function, …).

Given any small number $\epsilon > 0$ of an acceptable error, we can find a number *M* and weights $\mathbf{w}^{(2)} \in \mathbb{R}^M$ and $\mathbf{W}^{(1)} \in \mathbb{R}^{M \times D}$ such that:

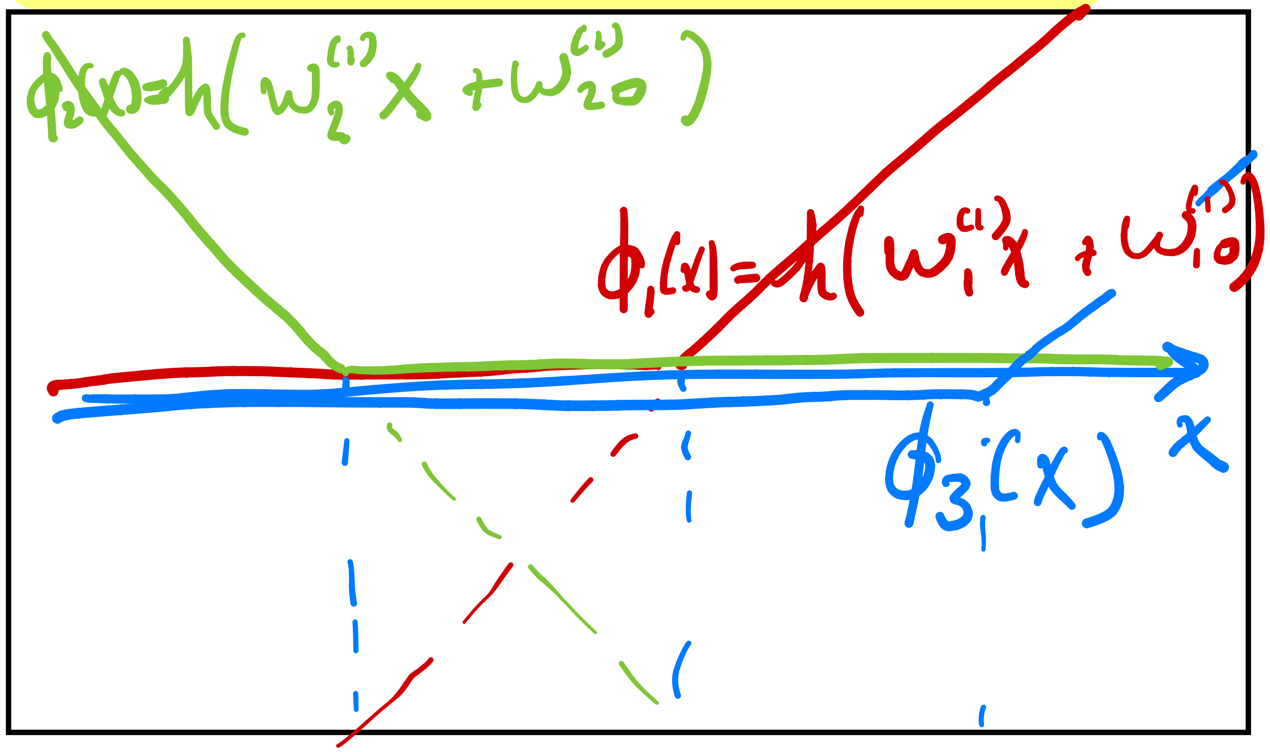
$$|f(\mathbf{x}) - y(\mathbf{x}, \mathbf{W}^{(1)}, \mathbf{w}^{(2)})| < \epsilon$$

with $y(\mathbf{x}, \mathbf{W}^{(1)}, \mathbf{w}^{(2)}) = \sum_{m=0}^{M} w_m^{(2)} \overbrace{h\left(\sum_{d=0}^{D} w_{md}^{(1)} x_d\right)}^{\phi_m(\underline{x})}$

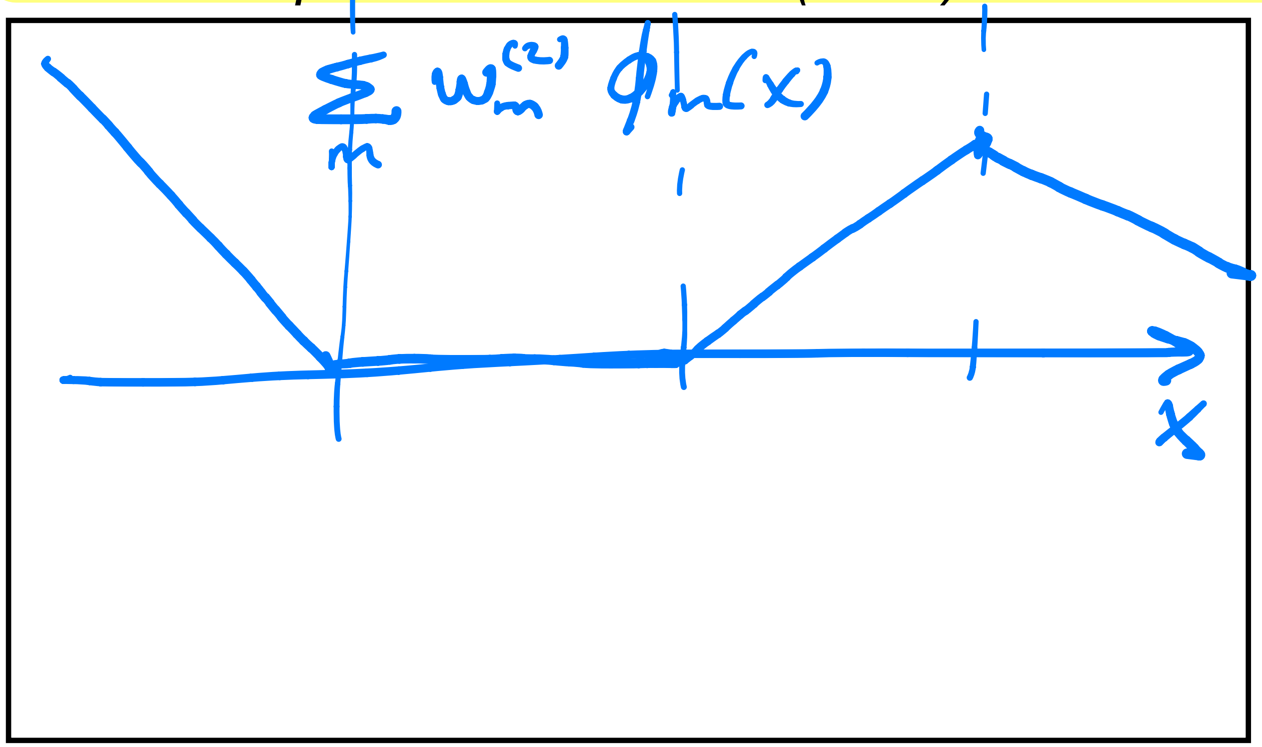
Caution: for smaller ϵ we usually need larger M

Neural Networks with ReLU $= \max(0, a)$

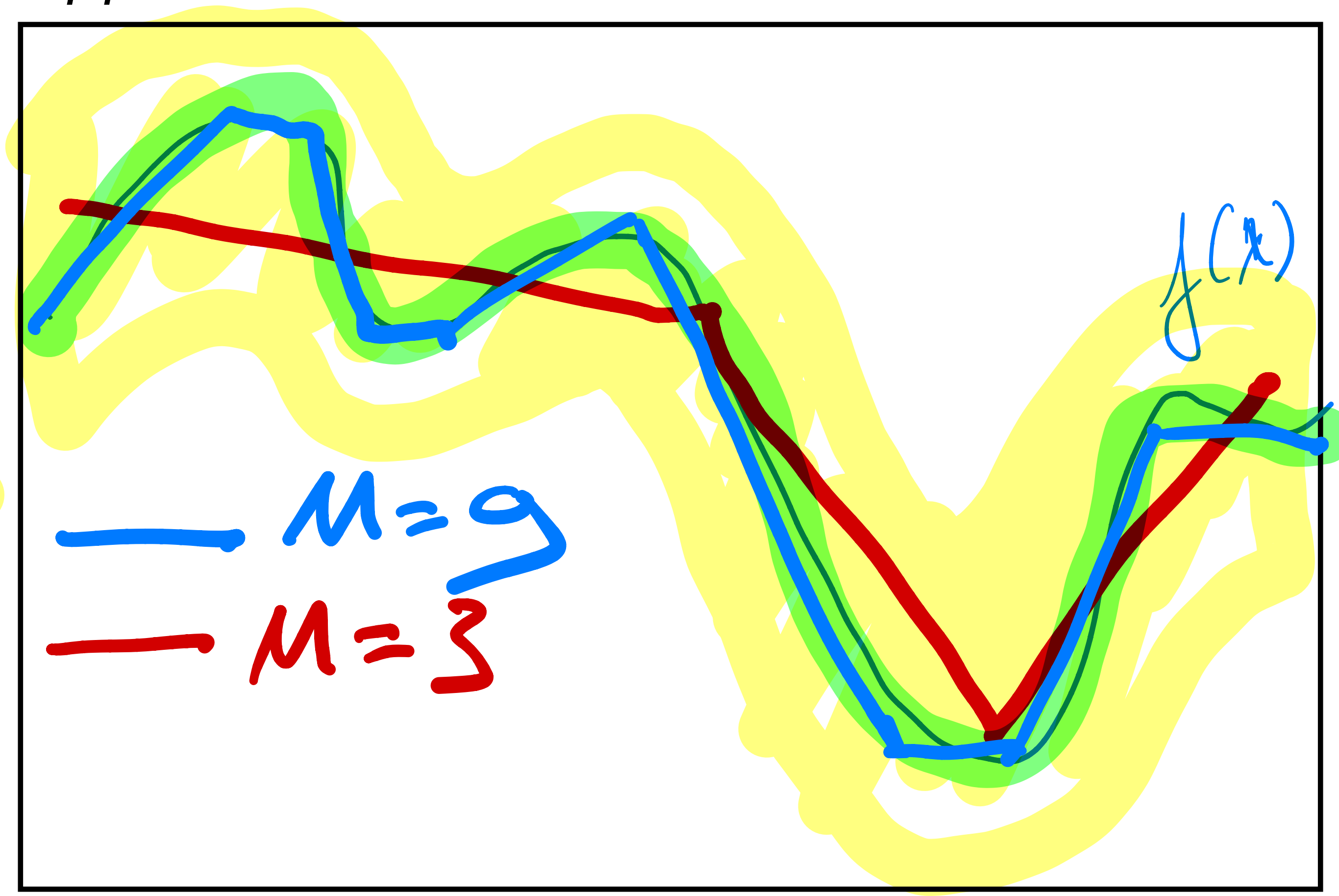
Learned basis functions with ReLU



Leads to piece-wise linear (PWL) functions



Approximation with ReLU NNs/PWL functions



Deep Neural Nets and Shallow Neural Nets

- Take a neural net with L layers.
- Take a more shallow neural net with L' < L layers.
- Approximate the deep neural net with shallow neural net up to error ε
- Usually number of units $M(\varepsilon)$ of shallow net scales exponentially for decreasing ε!

Expressive power ReLU networks

- Expressive power of ReLU-DNN = number of linear regions

$$\# \text{ regions} \approx \text{width}^{\text{depth} \cdot D}$$ ← input dim D

- Polynomial in width, but exponential with depth

- With fixed network capacity

$$\# \text{ parameters} \approx \text{width}^2 \cdot \text{depth}$$

- Most expressive power is gained by going deeper with less neurons per layer than staying shallow with more neurons per layer.

Example: Function Approximations

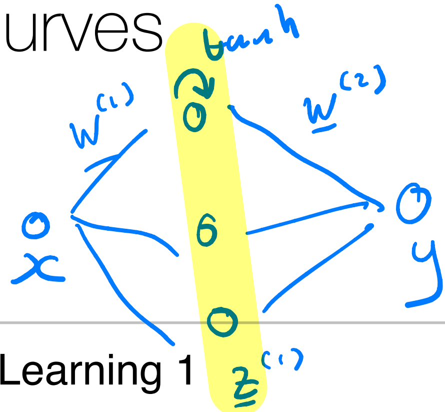
(a) $f(x) = x^2$

(b) $f(x) = \sin(x)$

(c) $f(x) = |x|$

(d) $f(x) = H(x)$

- ✦ N=50 datapoints
- ✦ MLP: 2 layers, 3 hidden units with tanh activation function. 1 linear output unit.
- ✦ Hidden unit outputs: dashed curves



$f(x) = x^2$

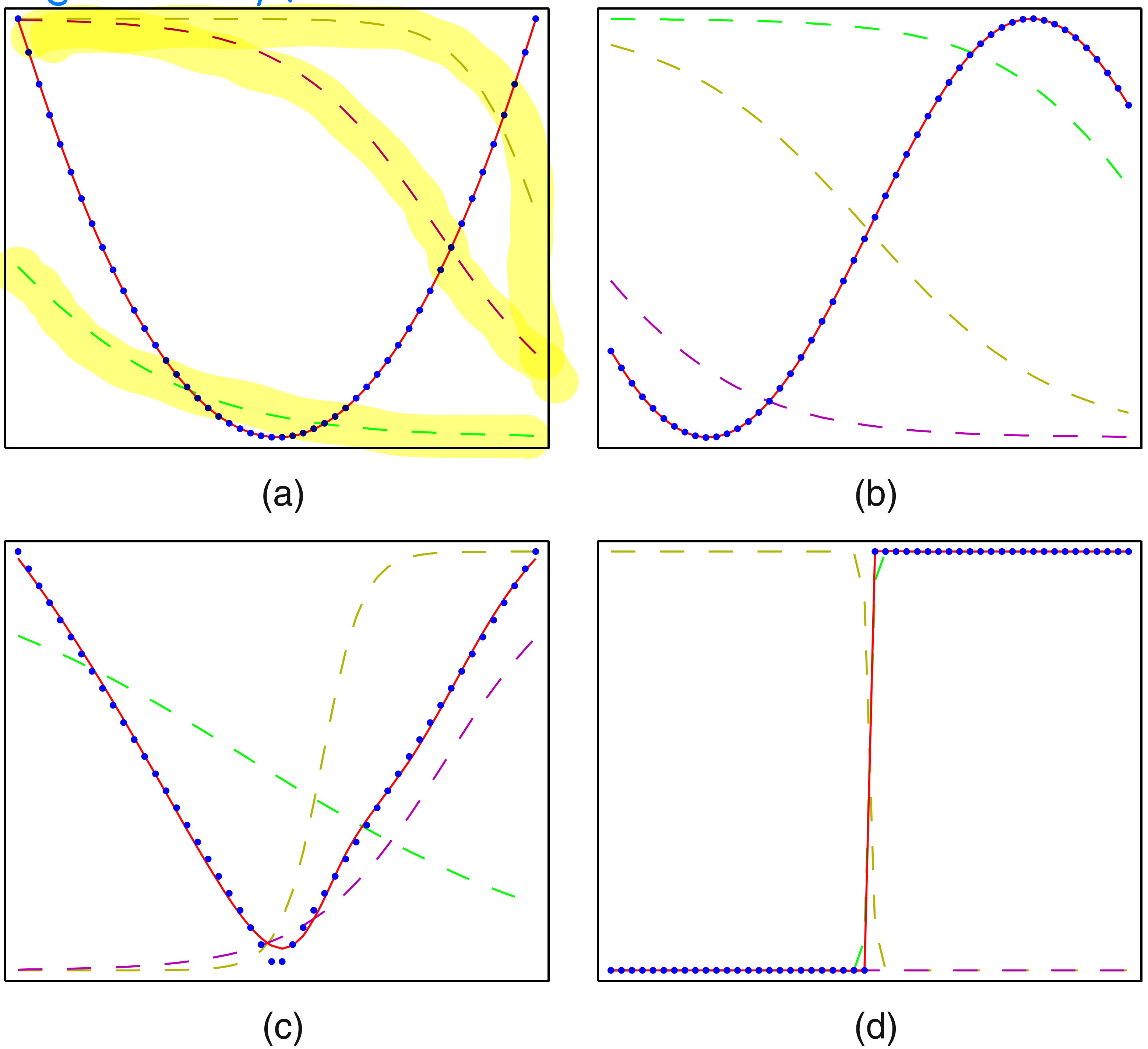
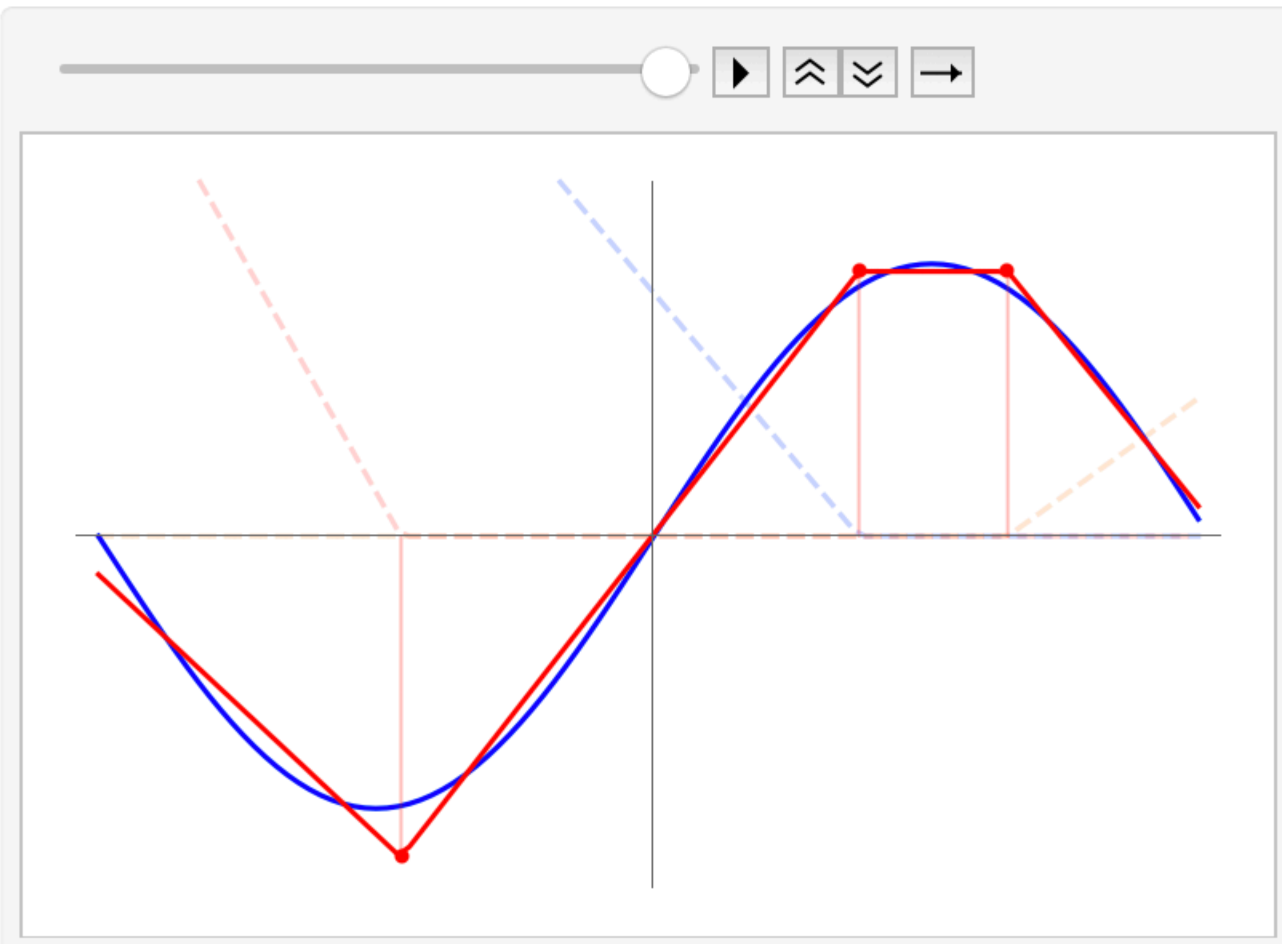


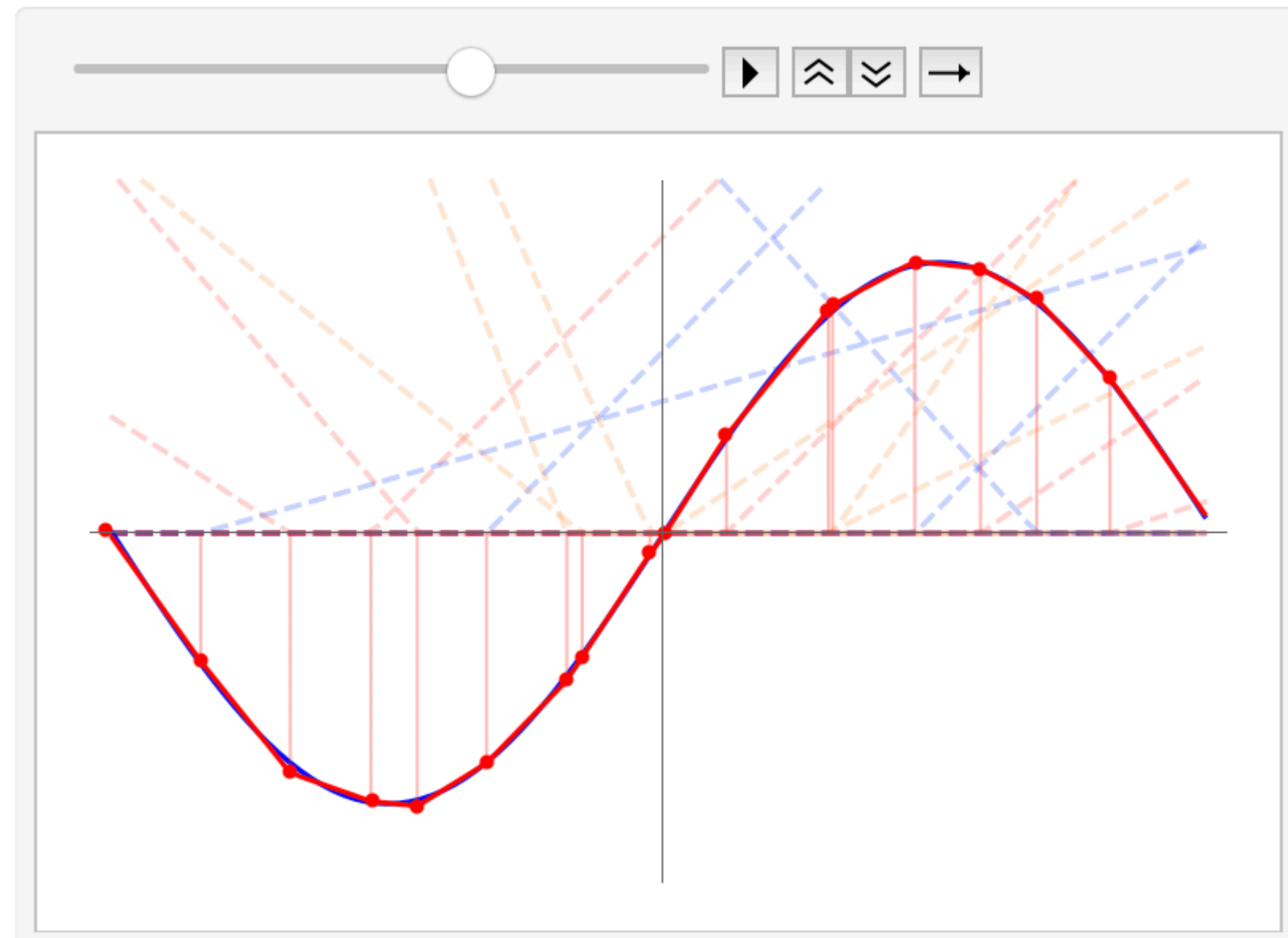
Figure: MLP approximating four different functions (red curves) (Bishop 5.3)

Example: Function Approximations

Piece-wise linear approximation with 2-layer NN with ReLU



M=3 hidden units



M=20 hidden units

Example: Classification with Neural Nets

Class 1 class 2

MLP:

- 2 layers
- # of inputs: 2
- 2 hidden units with tanh activation function
- # of outputs: 1
- Output activation function: σ : sigmoid
- Red line: MLP decision boundary
- Green line: optimal decision boundary from synthetic data distribution

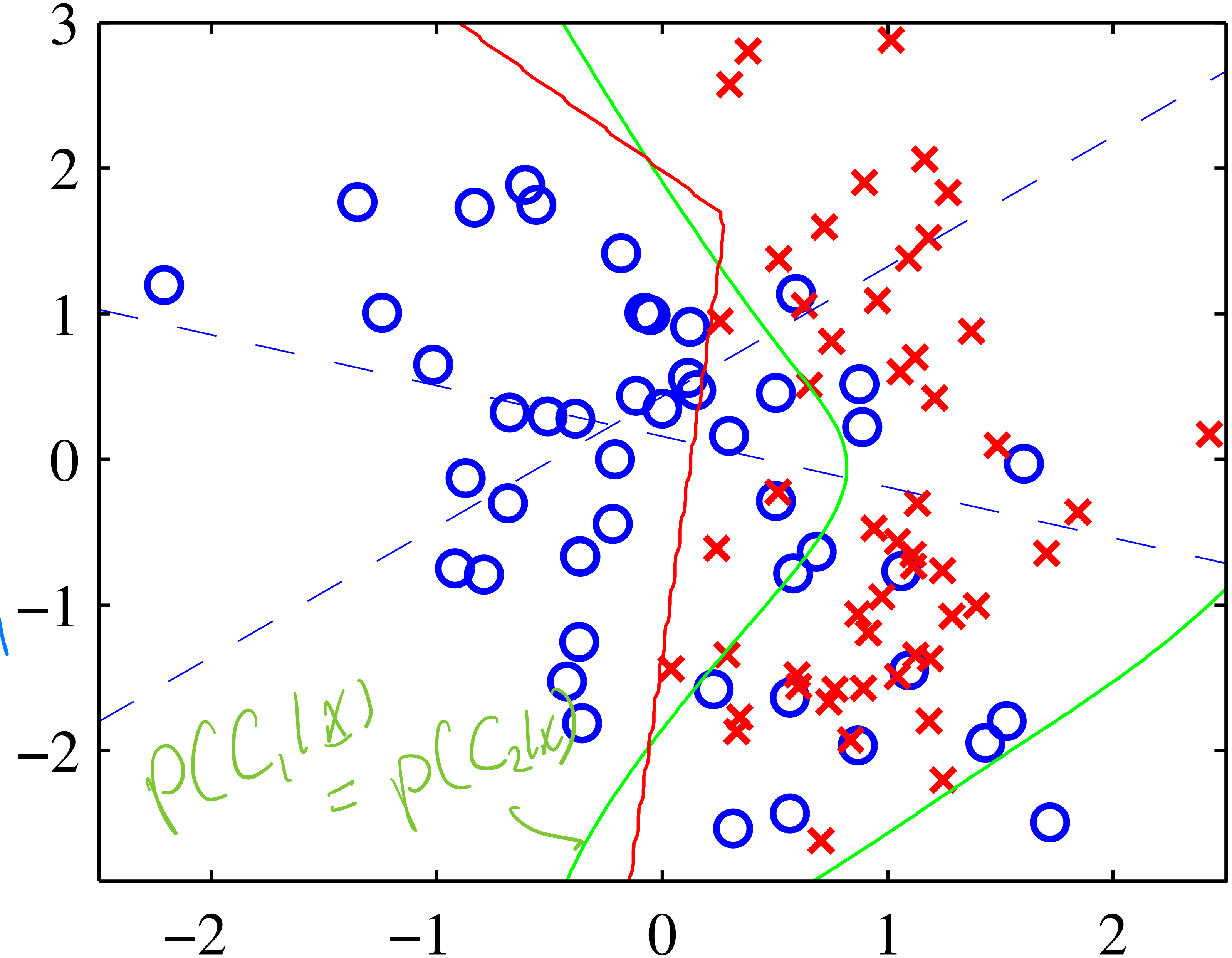


Figure: MLP for classification with 2 classes (Bishop5.4)